MODIS
SCIENCE DATA SUPPORT TEAM
PRESENTATION

December 20, 1991

AGENDA

1. Action Items
2. MODIS Airborne Simulator (MAS)
3. TEG Status
4. MODIS Team Leader Computing Facility Plan
5. Strawman SDST Work Plan Schedule

ACTION ITEMS:

08/30/91 [Lloyd Carpenter and Team]: Draft a schedule of work for the next 12 months. Include primary events and milestones, documents to be produced, software development, MAS support, etc. (The work Plan has been entered into Microsoft Project. A strawman schedule is included in the handout.) STATUS: Open. Due date 09/27/91.

10/04/91 [Phil Ardanuy and Team]: Prepare questions for the project to characterize the spacecraft position and attitude knowledge and the MODIS pointing knowledge in a way that will facilitate the evaluation of methods such as image registration to meet the science team requirements for earth location. (The letter to the project was prepared, 10/28/91.) STATUS: Open. Due date 10/18/91.

12/06/91 [Liam Gumley]: Investigate a cataloguing scheme for the MAS data. Consider the Master Catalogue, PLDS and PCDS. STATUS: Open. Due date 02/14/92.

12/06/91 [Liam Gumley, Tom Goff, Ed Masuoka]: Develop a plan for storing and distributing MAS data. STATUS: Open. Due date 02/14/92.

12/06/91 [Ed Masuoka]: Arrange for the "QA" company to come in and give a demonstration of their Performance Analysis Tool. (The demonstration took place at 10:00am Thursday, December 19, 1991.) STATUS: Closed. Due date 12/20/91.

MODIS Airborne Simulator status (Liam Gumley)

Progress up to 19 December 1991

(1) MAS Level-1B data delivery

An 8mm Exabyte tape containing 17 processed MAS flight tracks was delivered to Mike King on 12/18/91. A summary of the files included on the tape is shown overleaf. Each file represents a flight track, and the data is from the MAS flights on 11/12/91 and 11/14/91. Mike is satisfied that this amount of data will keep his programmers busy over the holiday break.

(2) MAS anonymous FTP site

The anonymous FTP site became operational on 12/17/91 on LTPIRIS2. Programmers for Mike King, John Barker, and Dorothy Hall have been notified of it's existence. An introduction to the files at this site has been appended.

(3) MAS visible/near-IR calibration

A new set of MAS visible/near-IR calibration coefficients was received from Tom Arnold on 12/12/91. These values differ by a few percent from the original numbers. In consultation with Tom it was decided to proceed with processing using the original set of numbers, and to re-process when a final set of accepted numbers is released. It is expected that a strategy for averaging/smoothing the data in the IR channels will also have been decided by this time.

(4) MAS data processing instructions / operator guide

Tom Goff has been run through the steps necessary to process MAS software on the LTP Iris and VAX should the need arise. A written set of instructions is under development.

MAS Level-1B Exabyte tape
18 December 1991
Liam Gumley, RDC (301 982 3748)

The accompanying Exabyte tape was created on a VAX/VMS system, and contains the following MAS netCDF files.

```
-rw-r--r--   1 gumley   staff    49953200 Dec 12 13:22 12nov91-01.cdf
-rw-r--r--   1 gumley   staff    86440800 Dec 16 18:17 14nov91-01.cdf
-rw-r--r--   1 gumley   staff    29981040 Dec 16 18:22 14nov91-02.cdf
-rw-r--r--   1 gumley   staff    53601960 Dec 16 18:47 14nov91-03.cdf
-rw-r--r--   1 gumley   staff    59363160 Dec 16 19:15 14nov91-04.cdf
-rw-r--r--   1 gumley   staff    49569120 Dec 16 19:38 14nov91-05.cdf
-rw-r--r--   1 gumley   staff    57634800 Dec 16 20:05 14nov91-06.cdf
-rw-r--r--   1 gumley   staff    43807920 Dec 16 20:25 14nov91-07.cdf
-rw-r--r--   1 gumley   staff    51873600 Dec 16 20:49 14nov91-08.cdf
-rw-r--r--   1 gumley   staff    54178080 Dec 16 21:14 14nov91-09.cdf
-rw-r--r--   1 gumley   staff    34013880 Dec 16 21:30 14nov91-10.cdf
-rw-r--r--   1 gumley   staff    14425800 Dec 17 09:35 14nov91-11.cdf
-rw-r--r--   1 gumley   staff    42079560 Dec 17 09:55 14nov91-12.cdf
-rw-r--r--   1 gumley   staff    44384040 Dec 17 10:16 14nov91-13.cdf
-rw-r--r--   1 gumley   staff    33437760 Dec 17 10:40 14nov91-14.cdf
-rw-r--r--   1 gumley   staff    25372080 Dec 17 10:52 14nov91-15.cdf
-rw-r--r--   1 gumley   staff    63396000 Dec 17 11:22 14nov91-16.cdf
```

The tape is unlabeled, 2048 bytes per block, 4 x 512 byte records per block.

To read this tape on a VAX/VMS system, use the following commands.

```
allocate mua0: tape                        (use appropriate drive name)
mount/foreign/blocksize=2048/recordsize=512 tape
set magtape/rewind tape
copy/log tape: diskfile01.cdf              (use appropriate file name)
copy/log tape: diskfile02.cdf
.
.
.
.
set magtape/rewind tape
dismount tape
deallocate tape
```

The file sizes will be slightly larger than those listed above, since VMS pads out the last block of the tape file to contain 2048 bytes. This does not affect the use of the netCDF files, as they have length information embedded. These netCDF files are in the format described in the document 00readme.doc. This document is found at the anonymous FTP site ltpiris2.gsfc.nasa.gov, in directory pub/MAS.

Introduction to MAS Level-1B data (17 December 1991)

Liam E. Gumley
Research and Data Systems Corporation (RDC)

An anonymous FTP account has been set up to enable GSFC users of MAS data to become familiar with the structure of MAS Level-1B datasets. Questions should be directed to:

Liam E. Gumley (RDC 301-982-3748 gumley@ltp.gsfc.nasa.gov), or
Thomas E. Goff (RDC 301-982-3704 teg@ltp.gsfc.nasa.gov).

Accessing the anonymous FTP account

The files in the account have been set up specifically for Silicon Graphics Iris users. It is therefore advisable to use your own Iris as a base for retrieving files. To connect to the host, type

```
ftp ltpiris2
```

For a username, enter

```
anonymous
```

and for a password, enter your Internet ID, e.g.

```
fred@myiris.gsfc.nasa.gov
```

You should then change to the MAS directory by typing

```
cd pub/MAS
```

Files in the anonymous FTP directory

```
-rw-r--r--   1 gumley   staff      13351 Dec 17 11:15 00readme.doc
-rw-r--r--   1 gumley   staff   49953200 Dec 17 09:33 12nov91-01.cdf
-rw-r--r--   1 gumley   staff        990 Dec 17 15:52 config.asc
-rw-r--r--   1 gumley   staff      85716 Dec 17 09:30 libnetcdf.a
-rwxr-xr-x   1 gumley   staff     136048 Dec 17 09:30 ncdump
-rw-r--r--   1 gumley   staff       1532 Dec 13 11:02 netcdf.doc
-rw-r--r--   1 gumley   staff       8844 Dec 17 09:30 netcdf.h
-rwxr-xr-x   1 gumley   staff     132292 Dec 17 09:38 subset10
-rw-r--r--   1 gumley   staff       2846 Dec 17 16:07 subset10.c
```

A description of each of these files follows.

00readme.doc (ASCII)
The file you are now reading.

12nov91-01.cdf (BINARY)

This file is a MAS flight track in netCDF form. The flight occurred on 12 November 1991 and was a ferry flight from Ames Research Center to Houston Texas for FIRE. The flight track is mostly cloud free, and includes snow cover and open ground. The region covered is the Sierra Nevada mountains in California. The file contains 2601 scanlines of MAS data.

config.asc (ASCII)
The MAS instrument configuration file used in Level-1 processing (self explanatory).

libnetcdf.a (BINARY)
This file is an Iris object library containing the compiled C object modules for the netCDF version 1.17 code. This library must be linked with any application that calls netCDF routines. The library routines were compiled under Irix version 3.3.

ncdump (BINARY)
This file is an executable program for the Iris, running Irix version 3.3. ncdump is a netCDF application that is distributed along with the netCDF source code by UCAR. It's function is to dump the entire contents of a netCDF file in human-readable ASCII format. For a large netCDF data file such as a MAS flight track, this produces *many* pages of screen output. It is possible to dump only the header of the netCDF file by typing

```
ncdump -h file.cdf
```

netcdf.doc (ASCII)
This is a text file that explains how to obtain the netCDF source code from UCAR by FTP. These instructions are Iris specific. While a compiled netCDF library has already been provided (libnetcdf.a) the documentation included in the UCAR package is *essential* for software development, and the source code is the latest version (2.01).

netcdf.h (ASCII)
This is a C header file which contains definitions used by the netCDF library routines. This file must be present when compiling programs that call netCDF library routines. The file is from the UCAR netCDF source code, and the version is 1.17.

subset10 (BINARY)
This file is an executable program for the Iris, running Irix version 3.3. subset10 was written in C by Thomas E. Goff (RDC) to subsample MAS netCDF flight track files and produce 'flat' binary 16 bit images of MAS radiances in a selected channel. It is worth noting that MAS radiances in all channels are scaled to 16 bit integers by the transformation

```
intrad = NINT( 100.0 * REALRAD )
```

subset10.c (ASCII)
This file contains the C source code for the subset10 program. This program was written by Thomas E. Goff (RDC). It gives an example of how the netCDF library routines are called to access netCDF files. The user may wish to modify this code to retrieve other portions of the netCDF data file such as geolocation. To compile this program under Irix 3.3, type

```
cc -I. -o subset10 subset10.c -L. -lnetcdf
```

This command causes the C compiler to look for netcdf.h and libnetcdf.a in the current directory.

Structure of the MAS netCDF flight track files

The following text data was produced by dumping the header of the file 12nov91-01.cdf using ncdump (ncdump -h 12nov91-01.cdf).

(text from ncdump follows)

```
netcdf 12nov91-01 {
dimensions:
        Time = UNLIMITED ; // (2601 currently)
        NumberOfChannels = 12 ;
        NumberOfPixels = 716 ;
        HeaderLength = 1840 ;
        AnchorIndexSize = 73 ;

variables:
        char DataSetHeader(HeaderLength) ;
        short AnchorPtIndex(AnchorIndexSize) ;
        short DataFrameStatus(Time) ;
        long ScanLineCounter(Time) ;
        long ThumbWheelSwitches(Time) ;
        short ScanRate(Time) ;
        long GMTime(Time) ;
        short S-BendIndicator(Time) ;
        short AircraftRollCount(Time) ;
        long Year&DayOfYear(Time) ;
        short BlkBdy1Temperature(Time, NumberOfChannels) ;
        short BlkBdy2Temperature(Time, NumberOfChannels) ;
        short AmplifierGain(Time, NumberOfChannels) ;
        short BlkBdy1Counts(Time, NumberOfChannels) ;
        short BlkBdy2Counts(Time, NumberOfChannels) ;
        float CalibrationSlope(Time, NumberOfChannels) ;
        float CalibrationIntercept(Time, NumberOfChannels) ;
        float PixelLatitude(Time, AnchorIndexSize) ;
        float PixelLongitude(Time, AnchorIndexSize) ;
        float SensorZenithAngle(Time, AnchorIndexSize) ;
        float SensorAzimuthAngle(Time, AnchorIndexSize) ;
        float SolarZenithAngle(Time, AnchorIndexSize) ;
        float SolarAzimuthAngle(Time, AnchorIndexSize) ;
        float AircraftLatitude(Time) ;
        float AircraftLongitude(Time) ;
        float AircraftHeading(Time) ;
        float AircraftAltitide(Time) ;
        float AircraftPitch(Time) ;
        short CalibratedData(Time, NumberOfChannels, NumberOfPixels) ;
}
```

(text from ncdump ends)

This information summarizes the variables contained in the MAS flight track file. All MAS flight track files have this same format. A more detailed summary of the above printout follows.

netcdf 12nov91-01 {
This simply reports the format and name of the file dumped.

dimensions:
This section defines the dimensions of the variables contained in the data set.

Time = UNLIMITED ; // (2601 currently)
This dimension is the 'length' dimension of the data set. It is extendable in the sense that extra data can be added at the end of the data set. Physically this dimension corresponds to elapsed time, however it also corresponds to the incrementing of the scan line counter.

NumberOfChannels = 12 ;
This is the number of channels on the MAS.

NumberOfPixels = 716 ;
This is the number of pixels across a MAS scan.

HeaderLength = 1840 ;
This is the length of the ASCII data set header. This is composed of blocks of 80 characters, which contain no carriage return, line feed or similar record separators. Currently the header contains one line of descriptive text that is entered by the person who processed the data, followed by 22 lines which describe the MAS instrument configuration as used in processing. This text is self-documenting.

AnchorIndexSize = 73 ;
This is the number of geolocation anchor points per scan line. Each MAS scan line has geolocation data for every 10th pixel (to save space). The geolocation data is defined for pixel numbers
1, 10, 20, 30, 40,......680, 690, 700, 710 and 716
which makes a total of 73 geolocation anchor points.
It should be noted that pixels 1 to 358 are on the starboard (right) side of the aircraft, while pixels 359 to 716 are on the port (left) side of the aircraft.

variables:
This section defines the type and size of the variables in the output data set. The type definitions are from the C language, however the equivalent FORTRAN types are
char = CHARACTER,
short = INTEGER*2,
long = INTEGER*4,
float = REAL*4.

char DataSetHeader(HeaderLength) ;
ASCII header text.
This indicates which spectral bands were selected, which channels had 8 or 10 bit data, which channels were calibrated using the MAS blackbodies (IR channels), and slopes/intercepts for the visible/near-IR channels.

short AnchorPtIndex(AnchorIndexSize) ;
Pixel numbers for which geolocation information is defined.

short DataFrameStatus(Time) ;
(Level-0 MAS engineering data)
Zero indicates good data.

long ScanLineCounter(Time) ;
(Level-0 MAS engineering data)
Scan line count, increments by 1 for every scan line.

long ThumbWheelSwitches(Time) ;
(Level-0 MAS engineering data)
Data system thumbwheel switch settings.

short ScanRate(Time) ;
(Level-0 MAS engineering data)
Scan rate in scans per second (x 10, nearest integer).

long GMTime(Time) ;
(Level-0 MAS engineering data)
Greenwich Mean Time (HHMMSSS).

short S-BendIndicator(Time) ;
(Level-0 MAS engineering data)
S-bend indicator : 0=no S-bend, 1=S-bend.

short AircraftRollCount(Time) ;
(Level-0 MAS engineering data)
Aircraft roll count (signed integer, positive is right),
0.03 degrees per count.

long Year&DayOfYear(Time) ;
Year, month, day (YYYYMMDD).

short BlkBdy1Temperature(Time, NumberOfChannels) ;
(Level-0 MAS engineering data)
Black Body 1 (cold) thermal reference temperature
(degrees C x 100)

short BlkBdy2Temperature(Time, NumberOfChannels) ;
(Level-0 MAS engineering data)
Black Body 2 (hot) thermal reference temperature
(degrees C x 100)

short AmplifierGain(Time, NumberOfChannels) ;
(Level-0 MAS engineering data)
Instrument Gain (x 1000)

short BlkBdy1Counts(Time, NumberOfChannels) ;
(Level-0 MAS engineering data)
Black Body 1 (cold) radiance count

```
short BlkBdy2Counts(Time, NumberOfChannels) ;
(Level-0 MAS engineering data)
Black Body 2 (hot) radiance count

float CalibrationSlope(Time, NumberOfChannels) ;
Count to radiance calibration slope.
Radiance = (count*slope + intercept)/gain (visible/near-IR channels),
Radiance =  count*slope + intercept       (IR channels).
Radiance units are:
milliWatts per square centimeter per steradian per micron
for visible/near-IR channels (calibrated by integrating sphere),
milliWatts per square centimeter per steradian per wavenumber
for IR channels (calibrated using MAS blackbodies).

float CalibrationIntercept(Time, NumberOfChannels) ;
Count to radiance calibration intercept.
Units are the same as the calibration slope.

float PixelLatitude(Time, AnchorIndexSize) ;
Latitudes for pixels at geolocation anchor points.
Latitude ranges from -90 degrees at the South Pole to
+90 degrees at the North Pole.

float PixelLongitude(Time, AnchorIndexSize) ;
Longitudes for pixels at geolocation anchor points.
Longitude is zero at the Greenwich Meridian, and ranges
from -180 degrees (West) to +180 degrees (East).

float SensorZenithAngle(Time, AnchorIndexSize) ;
MAS sensor zenith angle for pixels at geolocation anchor points.
Defined as the zenith angle (degrees) of a vector from
the sensor to the pixel (nadir sensor zenith angle = 0).

float SensorAzimuthAngle(Time, AnchorIndexSize) ;
MAS sensor azimuth angle for pixels at geolocation anchor points.
Defined as the azimuth angle (degrees) clockwise from
North of a vector from the pixel to the sensor.

float SolarZenithAngle(Time, AnchorIndexSize) ;
Solar zenith angle for pixels at geolocation anchor points.
Defined as the zenith angle (degrees) of a vector from
the pixel to the Sun.

float SolarAzimuthAngle(Time, AnchorIndexSize) ;
Solar azimuth angle for pixels at geolocation anchor points.
Defined as the azimuth angle (degrees) clockwise from
North of a vector from the pixel to the Sun.

float AircraftLatitude(Time) ;
Aircraft subpoint latitude (degrees, derived from INS data).

float AircraftLongitude(Time) ;
Aircraft subpoint longitude (degrees, derived from INS data).
```

```
float AircraftHeading(Time) ;
Aircraft heading (degrees, derived from INS data).

float AircraftAltitide(Time) ;
Aircraft altitude (meters, derived from INS data).

float AircraftPitch(Time) ;
Aircraft pitch angle (degrees, derived from INS data).

short CalibratedData(Time, NumberOfChannels, NumberOfPixels) ;
Calibrated MAS radiances for all channels and all pixels
(x 100, nearest integer).
Radiance units are:
milliWatts per square centimeter per steradian per micron
for visible/near-IR channels (calibrated by integrating sphere),
milliWatts per square centimeter per steradian per wavenumber
for IR channels (calibrated using MAS blackbodies).
```

Tom Goff's Status
for
19 December, 1991

TGoff on GSFC mail, or teg@LTPIRIS2.GSFC.NASA.GOV

* **Code Portability** - a baptism by fire: As part of my efforts to port the NetCDF XDR source files to/from PC's and the Mac, I ran into the infamous record termination lack of standardization. In order to solve this problem I needed to know what the various computer operating systems use as a record terminator. I picked up a file dumping utility, written in C, that was supposed to dump the contents of files in hex, decimal, octal, EBCDIC, and ASCII. Unfortunately, this program as written opened the files in non-binary mode which stripped certain record terminator characters from the file to be dumped. So I bit the bullet and modified the program to read files in binary mode and to decode the file contents in mnemonic form, integer form, or floating point form in addition the defaults. An attempt to port this simple program to the VAX and IRIS machines as a useful utility for file snooping was then undertaken. Three days later, the port had not been completed, but much knowledge of the various machines was obtained. This program will be made to perform file offset and dump size specifications and should be available shortly.

* **Pro:QA** demo - This package is used to analyze programming source code (statically - before execution) to find as many potential bugs and digressions from accepted coding standards as possible. The package is user-configurable in selecting severity levels and user criteria via scripting language. The MAS processing code was analyzed by this package and found to be "exceptionally well commented" with "only a few problems". "The MAS program must have been developed with structured programming techniques in mind" (goto's excepted).

  Having Pro:QA or a similar resource available to SDST program developers would be very beneficial in eliminating 90% of common coding errors (Pro:QA statistic). I feel that a tool with this capability would be most useful in helping to solve the problem of porting code from one machine to another. The tool checks for and flags any non-standard or non-portable aspects of the code that can then be modified to make the code more portable. This tool could be used to inform and enforce coding standards for code to be acceptable to the SDST for incorporation into the ECS PGS processing system.

  This tool works under X-windows Rev 4 presently and is in the process of being encapsulated into HP's SoftBench CASE environment. This allows the tool to be used by many non-concurrent users with access to the X-window Pro:QA client via internet. Future encapsulation into SoftBench will allow interoperability between this tool and Upper CASE and/or Configuration Management tools and/or machine level dynamic debugging tools.

Near-Term Communications Requirements for the MODIS Team Leader SCF

Function	Remote Site	Environment	Protocol	Medium	Rate (Kbps)	Standard
Remote SDST Support	Local	X-Windows/MOTIF	CSLIP	Phone Lines (4)	14.4	V32bis, V42bis
Run CASE Tools	SCFs	X-Windows/MOTIF	CSLIP	Phone Lines (3)	14.4	V32bis, V42bis
			TCP/IP	Internet		
Project Management Tools	Local		TCP/IP	Goddard Network		
ECS Toolkit Evaluation (Beta Testing)						
ESN Toolkit	PGS			ESN		
SMC (CASE) Toolkit	PGS			ESN		
IMS Toolkit	PGS			ESN		
	Anywhere		TCP/IP	Internet		
	Anywhere			Phone Lines (1)	14.4	V32 bis, V42 bis
Team-Member-Defined Support Processing	SCF			Phone Lines (1)	14.4	V32 bis, V42 bis
Preliminary Data Cataloging and Distribution System	Anywhere			Phone Lines (1)	14.4	V32 bis, V42 bis
Integration with Version-0	DAAC				56	

Near-Term Communications Requirements for the MODIS Team Leader SCF

Function	User	Remote Site	Environment	Protocol	Medium	Rate (Kbps)
Run CASE Tools	SDST	Terminal Room	X-Windows/MOTIF	TCP/IP	Goddard Network	
	TMs	SCFs	X-Windows/MOTIF	TCP/IP	Internet	
	TMs	SCFs	X-Windows/MOTIF	SLIP/CSLIP	Phone Line (1)	14.4, V32 bis, V42 bis
ECS Toolkit Evaluation (Beta Testing)						
ESN Toolkit	SDST	PGS			ESN	
SMC (CASE) Toolkit	SDST	PGS			ESN	
IMS Toolkit	SDST	PGS			ESN	
	SDST	Anywhere		TCP/IP	Internet	
	SDST	Anywhere			Phone Line (1)	14.4, V32 bis, V42 bis
Team-Member-Defined Support/ Integration and Testing Support	TMs	SCFs	TELNET X-Windows/MOTIF	TCP/IP	Internet	
	TMs	SCFs	X-Windows/MOTIF	SLIP/CSLIP	Phone Line (1)	14.4 V32 bis, V42 bis
Integration with Version-0	SDST	DAAC	TELNET X-Windows/MOTIF			

Software Hierarchy for the Near-Term TLCF

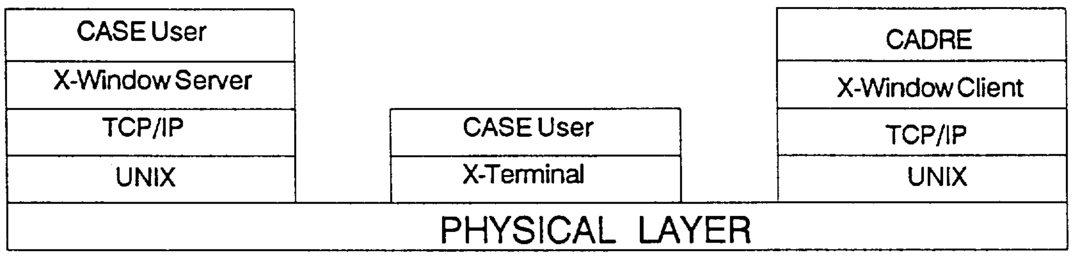

Figure 1. CASE Tool Operation

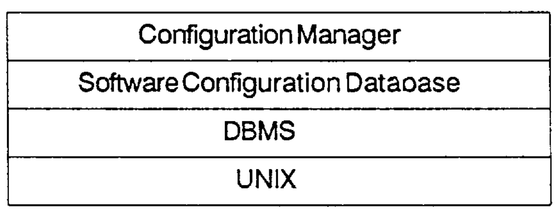

Figure 2. Software Configuration Management

Figure 3. ECS Toolkit Evaluation

Software Hierarchy for the Near-Term TLCF (continued)

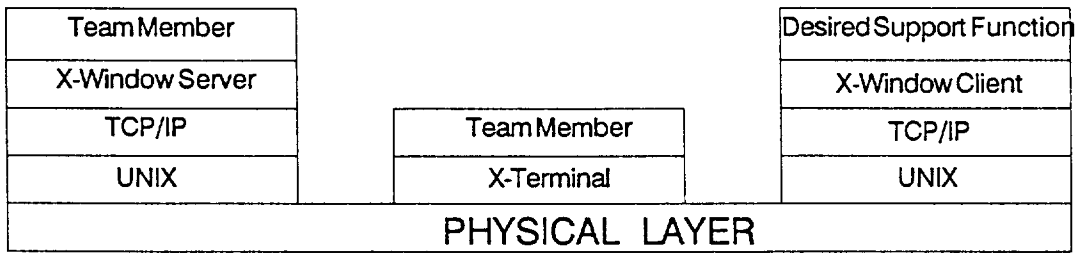

Figure 4. Team Member Support (image-based)

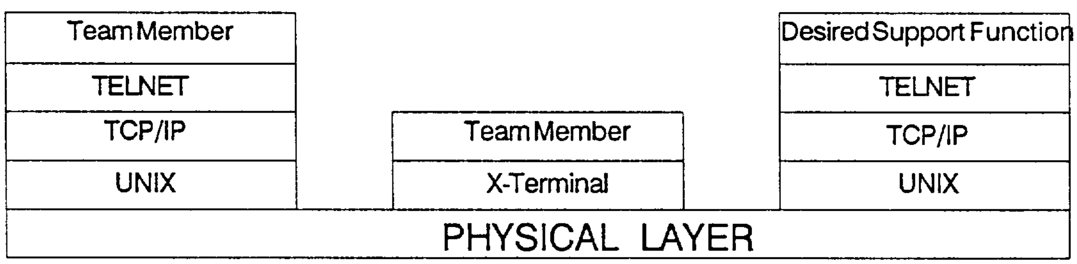

Figure 5. Team Member Support (character-based)

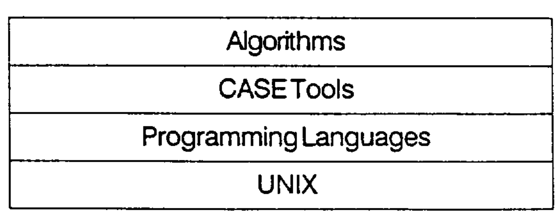

Figure 6. Algorithm Development and Product Generation

Some Notes on TLCF Communication Topics

1. The Internet should perhaps serve as the primary communications link between the TLCF and the Science Team Members (at their SCFs) until services of the EOSDIS Science Network (ESN) become available. Most of the MODIS Team Members have access to Internet. Six Team Members have not confirmed Internet access, either in the MODIS Science Team Member directory or with Barbara Conboy, who maintains a list of e-mail addresses for Team Members. One of the six (Phil Slater) was contacted, more or less at random, and it was confirmed that the Slater team does NOT have Internet access at this time (it may become available if they move to a different building next September).

Given the fact that Internet access is not universal among Team Members, and considering the need of experimenters to work under sometimes difficult field conditions, it appears that the TLCF should support alternative Team Member access, perhaps using common carrier telephone service.

2. Serial Line Internet Protocol (SLIP) provides a mechanism for the transfer of Ethernet information across a serial communications line, such as a telephone provides. SLIP provides complete Ethernet functionality but at a reduced transfer rate. Compressed SLIP, or CSLIP, involves packet header compression only, and reduces overhead associated with SLIP to about one-sixth its former value. The effect of compression is most apparent when only a small amount of data is being transmitted, such as when a program is operating in interactive mode. Large file transfers proceed nearly as fast using SLIP as using CSLIP.

The compression technique used with CSLIP involves transmitting only changes in header information; each header is compared with a reference header transmitted at the beginning of the transfer. Under very noisy conditions, the reference packet may have to be retransmitted, and CSLIP may become inefficient in a noisy environment. SLIP normally works better for large file transfers and noisy environments; CSLIP may improve response time for programs operated in interactive mode.

3. A new protocol that provides similar function (Point-to-Point Protocol, or PPP) has recently been introduced into the public domain. This protocol is reputed to provide improved serial throughput, especially in noisy environments.

4. "Telenet" (renamed Sprint International after a merger with Sprint) can provide required communication services (including those in Australia and France; Sprint provides service in Australia and compatible facilities are available). Both batch and interactive access modes are supported. The major expense is the Data Access Facility (DAF), which connects Sprint facilities to the TLCF computer. Cost is $900/mo. for 9.6 kbps support, $2,000/mo.

for 14.4 support, and $2,550/mo. for 19.2 kbps access. Fifty-six kbps service is also available. A $140/mo. account charge applies to all accounts. Connect charges are $6/hr. in major metropolitan areas (Chicago, Los Angeles) and $7.50/hr. for medium-sized and smaller areas (including most remote areas, 96% of the U.S. covered). "800" number support is available for nearly all areas of the U.S., so additional long distance charges do not apply. Rates are half after 9:00 PM daily, on weekends, and on holidays.

Our machine would need to support the X.25 protocol.

Strawman SDST Schedule

```
                                                1992                      1992
                                            Dec  Jan Feb Mar  Apr May Jun  Jul  Aug  Sep Oct
                                            +----+---+---+----+---+---+----+----+----+---+--
1  TM S/W DEVELOPMENT                          |                                         *
2  Draft List of Algorithms                 O.|.O------->.
3  Identify Steps                           O+X.......................................X
4  Estimate time for each step                 |         >----->.
5  Identify staffing implications           +  |  +   +         >--->.  +    +    +    +   +
6  Complete Schedule                           |                    >--->.
7  Identify Prototype Algorithms               |                        >--->.
8  SDST Prototype Action                       |                            >-----------X
9                                              |
10 CASE TOOLS                               +  |  +   +   +    +   +   +    +    +    +  *
11 CASE goals                               O.|...O--->.........................>
12 CASE candidates                             |      >>.........................>
13 CASE Purchase                               |       >->........................>
14 CASE Test Plan                              |         >---X........................X
15                                          +  |  +   +                                  +
16 S/W GUIDELINES                              |                                         *
17 EOSDIS Standards                         O.|...........O------------->..>>
18 S/W Language                                |                             >..>>->
19 Data Format                                 |                             >..>>->
20 Documentation                            +  |  +   +   +    +   +   +       >...>-----X
21                                             |
22 SDST ALGORITHMS                             |                                         *
23 Level-1A Steps, Identify                 O.|O--->
24 Schedule Time for Each Level-1a Step        | >..>--->
25 Level-1B Steps, Identify                 +  |     >..>---> +   +   +    +    +    +   +
26 Schedule Time for each Level-1B Step        |         >..>--->
27 Complete Schedule for SDST Algorithms       |             >..>---X..................X
28 Develop Algorithms (this year)              |             >..>---------------------X
29                                             |
30 MAS                                      +  |  +   +   +    +   +   +    +    +    +  *
31 Software Maintenance                     O.|.O-------------------------------------X
32 Level-1 Processing                       O+|------->.........................>
33 Field Support for MAS                    O.|..............O-X......................X
34 Data Users Guide                            |       >---X..........................X
35 Data Catalog                             +  |  +    >---X..........................X
36                                             |
37 FIRST TLCF PURCHASE                         |                                         *
38 Identify Hardware                        O>|
39 Procurement Plan                            >--->..........................>
40 Evolution Strategy                       +  |  >------X.........................X
41                                             |
42 PROJECT MANAGEMENT                          |                                         *
43 Schedules                                O.|O--->
44 Staffing                                    | >..>--->
45 Earned Value                             +  |     >..>---> +   +   +    +    +    +   +
46 Doc. Control                                |         >..>------->
47 Action Items                                |                 >..>----->
48 Assumptions                                 |                       >..>------>
49 Tracking List                               |                             >...>------X
50                                             |
51 TRAINING PLAN                               |                                         *
52 CASE Tools, Training                     O.|............O--------->
53 UNIX OS Training                            |       >..............>------->  +    +  +
54 Obj Oriented Language Training              |               >.............>---------X
55 TLCF Sys Management Training             O.|..........OX...........................X
56                                             |
57 DELIVERABLES                                |                                         *
58 Project Plan                             O.|O-------------------------------------X
59 S/W & Data Plan                          O.|O----------------------X..............X
60 TLCF Plan                                O.|O----------------------X..............X
61                                             |
62 SCHEDULE & RESOURCES                        |                                         *
63 Schedule                                 O.|O----------------->  +    +    +    +   +
64 Resources                                   |                   >====================X
                                            +----+---+---+----+---+---+----+----+----+---+
```

```
O   Start of an activity with no predecessors          ===  Critical Path
>   Start of an activity with predecessors or          ...  Slack time
    end of activity with successors                    ---  Non-critical path
X   End of an activity with no successors               |   Current date
*   Milestone (no duration)
```

Strawman Network of SDST Tasks

Strawman Network of SDST Tasks

63 Schedule | Dec 23, 1991 8 | May 14, 1992 5 | Duration:20.00 | Slack:3.00 Week
64 Resources | May 15, 1992 8 | Oct 1, 1992 5 | Duration:20.00 | Slack:0 Week

43 Schedules | Dec 23, 1991 8 | Jan 22, 1992 5 | Duration:4.00 | Slack:3.00 Week
44 Staffing | Jan 23, 1992 8 | Feb 20, 1992 5 | Duration:4.00 | Slack:3.00 Week
45 Earned Value | Feb 21, 1992 8 | Mar 19, 1992 5 | Duration:4.00 | Slack:3.00 Week
46 Doc. Control | Mar 20, 1992 8 | May 14, 1992 5 | Duration:8.00 | Slack:3.00 Week
47 Action Items | May 15, 1992 8 | Jun 25, 1992 5 | Duration:6.00 | Slack:3.00 Week
48 Assumptions | Jun 26, 1992 8 | Aug 6, 1992 5 | Duration:6.00 | Slack:3.00 Week
49 Tracking List | Aug 7, 1992 8 | Oct 1, 1992 5 | Duration:8.00 | Slack:3.00 Week

2 Draft List of A | Dec 30, 1991 8 | Feb 26, 1992 5 | Duration:8.00 | Slack:4.30 Week
4 Estimate time f | Feb 27, 1992 8 | Apr 8, 1992 5 | Duration:6.00 | Slack:0.50 Week
5 Identify staffi | Apr 9, 1992 8 | May 6, 1992 5 | Duration:4.00 | Slack:0.50 Week
6 Complete Schedu | May 7, 1992 8 | Jun 3, 1992 5 | Duration:4.00 | Slack:0.50 Week
7 Identify Protot | Jun 4, 1992 8 | Jul 1, 1992 5 | Duration:4.00 | Slack:0.50 Week
8 SDST Prototype | Jul 2, 1992 8 | Sep 29, 1992 11 | Duration:12.70 | Slack:0.50 Week

23 Level-1A Steps. | Dec 23, 1991 8 | Jan 22, 1992 5 | Duration:4.00 | Slack:3.00 Week
24 Schedule Time f | Jan 23, 1992 8 | Feb 20, 1992 5 | Duration:4.00 | Slack:3.00 Week
25 Level-1B Steps. | Feb 21, 1992 8 | Mar 19, 1992 5 | Duration:4.00 | Slack:3.00 Week
26 Schedule Time f | Mar 20, 1992 8 | Apr 16, 1992 5 | Duration:4.00 | Slack:3.00 Week
27 Complete Schedu | Apr 17, 1992 8 | May 14, 1992 5 | Duration:4.00 | Slack:23.00 Wee
28 Develop Algorit | Apr 17, 1992 8 | Oct 1, 1992 5 | Duration:24.00 | Slack:3.00 Week

11 CASE goals | Jan 14, 1992 8 | Feb 11, 1992 5 | Duration:4.00 | Slack:32.00 Wee
12 CASE candidates | Feb 12, 1992 8 | Feb 19, 1992 5 | Duration:1.00 | Slack:26.20 Wee
13 CASE Purchase | Feb 20, 1992 8 | Mar 4, 1992 5 | Duration:2.00 | Slack:26.20 Wee
14 CASE Test Plan | Mar 5, 1992 8 | Apr 1, 1992 5 | Duration:4.00 | Slack:26.20 Wee

17 EOSDIS Standard | Mar 16, 1992 8 | Jul 3, 1992 5 | Duration:16.00 | Slack:16.00 Wee
18 S/W Language | Jul 31, 1992 8 | Aug 13, 1992 5 | Duration:2.00 | Slack:3.80 Week
20 Documentation | Aug 14, 1992 8 | Oct 1, 1992 5 | Duration:7.00 | Slack:3.80 Week

38 Identify Hardwa | Dec 2, 1991 8 | Dec 13, 1991 5 | Duration:2.00 | Slack:29.00 Wee
39 Procurement Pla | Dec 16, 1991 8 | Jan 14, 1992 5 | Duration:4.00 | Slack:29.00 Wee
40 Evolution Strat | Jan 15, 1992 8 | Mar 12, 1992 5 | Duration:8.00 | Slack:29.00 Wee

52 CASE Tools, Tra | Mar 20, 1992 8 | May 28, 1992 5 | Duration:10.00 | Slack:15.00 Wee
53 UNIX OS Trainin | May 29, 1992 8 | Jul 23, 1992 5 | Duration:8.00 | Slack:15.00 Wee
54 Obj Oriented La | Jul 24, 1992 8 | Oct 1, 1992 5 | Duration:10.00 | Slack:15.00 Wee

19 Data Format | Jul 31, 1992 8 | Aug 13, 1992 5 | Duration:2.00 | Slack:3.80 Week

32 Level-1 Process | Dec 2, 1991 8 | Feb 12, 1992 5 | Duration:10.00 | Slack:29.00 Wee
34 Data Users Guid | Feb 13, 1992 8 | Mar 12, 1992 5 | Duration:4.00 | Slack:29.00 Wee
35 Data Catalog | Feb 13, 1992 8 | Mar 12, 1992 5 | Duration:4.00 | Slack:29.00 Wee

1 TM S/W DEVELOPM | Oct 1, 1992 5 | Oct 1, 1992 5 | Duration:0.00 | Slack:215.00 Da

3 Identify Steps | Dec 2, 1991 8 | Dec 20, 1991 5 | Duration:3.00 | Slack:40.00 Wee